UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/606,641	06/29/2000	Kimberly J. Rush	MICR0512	7703

27792   7590   12/06/2004

MICROSOFT CORPORATION
LAW OFFICES OF RONALD M. ANDERSON
600 108TH AVENUE N.E., SUITE 507
BELLEVUE, WA 98004

EXAMINER
CHOUDHURY, AZIZUL Q

ART UNIT	PAPER NUMBER
2145	

DATE MAILED: 12/06/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Status

1)☒ Responsive to communication(s) filed on _18 August 2004_.

2a)☐ This action is **FINAL**.   2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is
closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _21-50_ is/are pending in the application.

     4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _21-50_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on _29 June 2000_ is/are: a)☒ accepted or b)☐ objected to by the Examiner.

     Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

     Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

     a)☐ All   b)☐ Some * c)☐ None of:

        1.☐ Certified copies of the priority documents have been received.

        2.☐ Certified copies of the priority documents have been received in Application No. _____.

        3.☐ Copies of the certified copies of the priority documents have been received in this National Stage
application from the International Bureau (PCT Rule 17.2(a)).

     * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☒ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4)☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5)☐ Notice of Informal Patent Application (PTO-152)

6)☐ Other: _____.

Detailed Action

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set
forth in section 102 of this title, if the differences between the subject matter sought to be patented and
the prior art are such that the subject matter as a whole would have been obvious at the time the
invention was made to a person having ordinary skill in the art to which said subject matter pertains.
Patentability shall not be negatived by the manner in which the invention was made.

Claims 21-50 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Doyle et al (US Pat No: 5838906) in view of Pratt (US Pat No: 5564044), hereafter

referred to as Doyle and Pratt, respectively.

1. With regards to claim 21, Doyle teaches through Pratt, the method for accessing

   multiple types of electronic content, comprising: receiving a request for a

   computer program to process an input to obtain an output comprising a type of

   content that is unknown to the computer program, wherein a service manager

   connects the computer program to at least one service container to process the

   input to obtain the output; selecting at least one segment of computer code from

   a plurality of segments of computer code that will enable the computer program

   to process the input, when the at least one segment of computer code is

   executed along with the computer program, to provide the output comprising the

   type of content that is unknown to the computer program; and executing the at

   least one segment of computer code along with the computer program to process

the input and obtain the output comprising the type of content that is unknown to

the computer program, wherein the plurality of segments of computer code and

the at least one segment of computer code are not executable as an independent

computer program

(Doyle teaches a design that features a "list of applications," (column 15,

line 14, Doyle). The list of applications allows an application to be selected to run

along with the original program to provide an output from an input that the

original program alone would be unable to provide as claimed. In addition, since

the list of applications is initiated by the system, there must exist a service

manager as claimed. However, such means are not expressly stated within

Doyle's design.

Pratt discloses a design allowing a second program to be called to assist a

first program to perform a desired task on data (column 3, lines 11-20, Pratt).

Within the disclosure, Pratt states that the system resolves the reference to

identify the script, the application program and the operation of the application

program to execute the script and invoke the application program to execute the

script to generate the result (column 3, lines 21-43, Pratt). Hence the claimed

service manager must be present within the design since it's tasks are

performed.

Both Doyle and Pratt teach systems allowing one program to call upon

another program to perform an operation on a piece of data. Hence, it would

have been obvious to one skilled in the art, during the time of the invention, to

have combined the teachings of Doyle with those of Pratt, to provide a

mechanism and a method for integrating first data created by a first application

program and residing in a first data file or object belonging to the first application

into a second data file or object (column 3, lines 11-15, Pratt)).

2. With regards to claim 22, Doyle teaches through Pratt, a method wherein selecting

at least one segment of computer code comprises selecting at least two

segments of computer code from the plurality of segments of computer code

whose combined functionality will enable the computer program to process the

input, when the at least two segments of computer code are executed along with

the computer program, to provide the output comprising the type of content that

is unknown to the computer program

(Doyle teaches a design that features a "list of applications," (column 15,

line 14, Doyle). The list of applications allows an application to be selected to run

along with the original program to provide an output from an input that the

original program alone would be unable to provide, as claimed. No limitation is

given as to how many applications may be selected to run together, hence two or

more may be selected as claimed. In addition, since the list of applications is

initiated by the system, there must exist a service manager as claimed.

However, such means are not expressly stated within Doyle's design.

Pratt discloses a design allowing a second program to be called to assist a

first program to perform a desired task on data (column 3, lines 11-20, Pratt).

Within the disclosure, Pratt states that the system resolves the reference to identify the script, the application program and the operation of the application program to execute the script and invoke the application program to execute the script to generate the result (column 3, lines 21-43, Pratt). Pratt's design also places no limitations on the number of applications that may be run along with the original program. Hence the claimed service manager must be present within the design since its tasks are performed.

Both Doyle and Pratt teach systems allowing one program to call upon another program to perform an operation on a piece of data. Hence, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Doyle with those of Pratt, to provide a mechanism and a method for integrating first data created by a first application program and residing in a first data file or object belonging to the first application into a second data file or object (column 3, lines 11-15, Pratt)).

3. With regards to claim 23, Doyle teaches through Pratt, the method further comprising configuring the at least two segments of computer code to be executed along with the computer program in a particular order to provide a desired processing of the input

(Doyle teaches a design that features a "list of applications," (column 15, line 14, Doyle). The list of applications allows an application to be selected to run along with the original program to provide an output from an input that the

original program alone would be unable to provide, as claimed. No limitation is given as to how many applications may be selected to run together; hence two or more may be selected as claimed. Whenever multiple applications run together to produce a single output, it is inherent that they will perform in order as claimed. In addition, since the list of applications is initiated by the system, there must exist a service manager as claimed. However, such means are not expressly stated within Doyle's design.

Pratt discloses a design allowing a second program to be called to assist a first program to perform a desired task on data (column 3, lines 11-20, Pratt). Within the disclosure, Pratt states that the system resolves the reference to identify the script, the application program and the operation of the application program to execute the script and invoke the application program to execute the script to generate the result (column 3, lines 21-43, Pratt). Hence the claimed service manager must be present within the design since it's tasks are performed.

Both Doyle and Pratt teach systems allowing one program to call upon another program to perform an operation on a piece of data. Hence, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Doyle with those of Pratt, to provide a mechanism and a method for integrating first data created by a first application program and residing in a first data file or object belonging to the first application into a second data file or object (column 3, lines 11-15, Pratt)).

4. With regards to claim 24, Doyle teaches through Pratt, a method further

comprising configuring the at least two segments of computer code into a

master-slave relationship that causes the execution of one of the at least two

segments of computer code to be dependent on the execution of another of the

at least two segments of computer code

(Doyle discloses how applications can be started as child processes

(column 15, line 22, Doyle). When a child process exists, a parent process must

exist. This parent-child relationship is equivalent to a master-slave relationship.

In addition, since the list of applications is initiated by the system, there must

exist a service manager as claimed. However, such means are not expressly

stated within Doyle's design.

Pratt discloses a design allowing a second program to be called to assist a

first program to perform a desired task on data (column 3, lines 11-20, Pratt).

Within the disclosure, Pratt states that the system resolves the reference to

identify the script, the application program and the operation of the application

program to execute the script and invoke the application program to execute the

script to generate the result (column 3, lines 21-43, Pratt). Hence the claimed

service manager must be present within the design since it's tasks are

performed.

Both Doyle and Pratt teach systems allowing one program to call upon

another program to perform an operation on a piece of data. Hence, it would

have been obvious to one skilled in the art, during the time of the invention, to

have combined the teachings of Doyle with those of Pratt, to provide a

mechanism and a method for integrating first data created by a first application

program and residing in a first data file or object belonging to the first application

into a second data file or object (column 3, lines 11-15, Pratt)).

5. With regards to claim 25, Doyle teaches through Pratt, a method wherein

executing the at least one segment of computer code comprises integrating the

at least one segment of computer code into the computer program and executing

the computer program to process the input and obtain the output comprising the

type of content that is unknown to the computer program

    (Doyle teaches a design that features a "list of applications," (column 15,

line 14, Doyle). This list of applications allows an application to be selected to

run along with the original program to provide an output from an input that the

original program alone would be unable to provide. When an application runs

along with the original program, Doyle states that it is "embedded" (column 6,,

lines 58-59, Doyle). Embedded is equivalent to the claimed integrated. In

addition, since the list of applications is initiated by the system, there must exist a

service manager as claimed. However, such means are not expressly stated

within Doyle's design.

Pratt discloses a design allowing a second program to be called to assist a first

program to perform a desired task on data (column 3, lines 11-20, Pratt). Within

the disclosure, Pratt states that the system resolves the reference to identify the

script, the application program and the operation of the application program to

execute the script and invoke the application program to execute the script to

generate the result (column 3, lines 21-43, Pratt). Hence the claimed service

manager must be present within the design since it's tasks are performed. Plus,

when a second program runs along with the first program in Pratt's design, they

are said to be integrated (column 3, lines 11-13, Pratt).

Both Doyle and Pratt teach systems allowing one program to call upon

another program to perform an operation on a piece of data. Hence, it would have

been obvious to one skilled in the art, during the time of the invention, to have

combined the teachings of Doyle with those of Pratt, to provide a mechanism and

a method for integrating first data created by a first application program and

residing in a first data file or object belonging to the first application into a second

data file or object (column 3, lines 11-15, Pratt)).

6. With regards to claim 26, Doyle teaches through Pratt, a method wherein receiving

a request for a computer program to process an input comprises receiving a

command to translate a word from a first language to a second language

(Doyle discloses a design that features a "list of applications," (column 15,

line 14, Doyle). These applications are able to run along with an original program

to provide an output from an input the original program is unable to produce

alone. No limitation is placed on the type of programs available, hence

translation programs are usable within the scope of the design. In addition, since the list of applications is initiated by the system, there must exist a service manager as claimed. However, such means are not expressly stated within Doyle's design.

Pratt discloses a design allowing a second program to be called to assist a first program to perform a desired task on data (column 3, lines 11-20, Pratt). Within the disclosure, Pratt states that the system resolves the reference to identify the script, the application program and the operation of the application program to execute the script and invoke the application program to execute the script to generate the result (column 3, lines 21-43, Pratt). Hence the claimed service manager must be present within the design since it's tasks are performed. Plus, Pratt's design also places no limitation on the types of programs that are usable, hence language translation programs are within the scope of it's design as well.

Both Doyle and Pratt teach systems allowing one program to call upon another program to perform an operation on a piece of data. Hence, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Doyle with those of Pratt, to provide a mechanism and a method for integrating first data created by a first application program and residing in a first data file or object belonging to the first application into a second data file or object (column 3, lines 11-15, Pratt)).

7. With regards to claim 27, Doyle teaches through Pratt, a method wherein receiving a request for a computer program to process an input comprises receiving a command to convert a number from a first number format to a second number format

(Doyle discloses a design that features a "list of applications," (column 15, line 14, Doyle). These applications are able to run along with an original program to provide an output from an input the original program is unable to produce alone. No limitation is placed on the type of programs available, hence number conversion programs are usable within the scope of the design. In addition, since the list of applications is initiated by the system, there must exist a service manager as claimed. However, such means are not expressly stated within Doyle's design.

Pratt discloses a design allowing a second program to be called to assist a first program to perform a desired task on data (column 3, lines 11-20, Pratt). Within the disclosure, Pratt states that the system resolves the reference to identify the script, the application program and the operation of the application program to execute the script and invoke the application program to execute the script to generate the result (column 3, lines 21-43, Pratt). Hence the claimed service manager must be present within the design since it's tasks are performed. Plus, Pratt's design also places no limitation on the types of programs that are usable, hence number conversion programs are within the scope of it's design as well.

Both Doyle and Pratt teach systems allowing one program to call upon

another program to perform an operation on a piece of data. Hence, it would

have been obvious to one skilled in the art, during the time of the invention, to

have combined the teachings of Doyle with those of Pratt, to provide a

mechanism and a method for integrating first data created by a first application

program and residing in a first data file or object belonging to the first application

into a second data file or object (column 3, lines 11-15, Pratt)).

8. With regards to claim 28, Doyle teaches through Pratt, the method wherein

receiving a request for a computer program to process an input comprises

receiving a command to convert a text object from a first text format to a second

text format

(Doyle discloses a design that features a "list of applications," (column 15,

line 14, Doyle). These applications are able to run along with an original program

to provide an output from an input the original program is unable to produce

alone. No limitation is placed on the type of programs available, hence text

format conversion programs are usable within the scope of the design. In

addition, since the list of applications is initiated by the system, there must exist a

service manager as claimed. However, such means are not expressly stated

within Doyle's design.

Pratt discloses a design allowing a second program to be called to assist a

first program to perform a desired task on data (column 3, lines 11-20, Pratt).

Within the disclosure, Pratt states that the system resolves the reference to
identify the script, the application program and the operation of the application
program to execute the script and invoke the application program to execute the
script to generate the result (column 3, lines 21-43, Pratt). Hence the claimed
service manager must be present within the design since it's tasks are
performed. Plus, Pratt's design also places no limitation on the types of
programs that are usable; hence text format conversion programs are within the
scope of its design as well.

Both Doyle and Pratt teach systems allowing one program to call upon
another program to perform an operation on a piece of data. Hence, it would
have been obvious to one skilled in the art, during the time of the invention, to
have combined the teachings of Doyle with those of Pratt, to provide a
mechanism and a method for integrating first data created by a first application
program and residing in a first data file or object belonging to the first application
into a second data file or object (column 3, lines 11-15, Pratt)).

9. With regards to claim 29, Doyle teaches through Pratt, a method wherein
receiving a request for a computer program to process an input comprises
receiving a command to convert a graphical object from a first graphical format to
a second graphical format

(Doyle discloses a design that features a "list of applications," (column 15,
line 14, Doyle). These applications are able to run along with an original program

to provide an output from an input the original program is unable to produce alone. No limitation is placed on the type of programs available; hence graphical format conversion programs are usable within the scope of the design. In addition, since the list of applications is initiated by the system, there must exist a service manager as claimed. However, such means are not expressly stated within Doyle's design.

Pratt discloses a design allowing a second program to be called to assist a first program to perform a desired task on data (column 3, lines 11-20, Pratt). Within the disclosure, Pratt states that the system resolves the reference to identify the script, the application program and the operation of the application program to execute the script and invoke the application program to execute the script to generate the result (column 3, lines 21-43, Pratt). Hence the claimed service manager must be present within the design since it's tasks are performed. Plus, Pratt's design also places no limitation on the types of programs that are usable; hence graphical format conversion programs are within the scope of its design as well.

Both Doyle and Pratt teach systems allowing one program to call upon another program to perform an operation on a piece of data. Hence, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Doyle with those of Pratt, to provide a mechanism and a method for integrating first data created by a first application

program and residing in a first data file or object belonging to the first application into a second data file or object (column 3, lines 11-15, Pratt)).

10. With regards to claim 30, Doyle teaches through Pratt, a computer system for accessing multiple types of electronic content, comprising: a processing unit; a memory in communication with the processing unit; and a computer program stored in the memory that provides instructions to the processing unit, wherein the processing unit is responsive to the instructions, operable for: identifying a plurality of segments of computer code that can be executed along with the computer program by the processing unit in response to the instructions; selecting, in response to an input command to access at least one type of content that the computer program is not configured to access, at least one segment of computer code from the plurality of segments of computer code that can be executed along with the computer program by the processing unit, in response to the instructions, to access the at least one type of content that the computer program is not configured to access, wherein a service manager connects the computer program to at least one service container in response to the input command; and executing the at least one segment of computer code along with the computer program to access the at least one type of content that the computer program is not configured to access; wherein the plurality of segments of computer code and the at least one segment of computer code are not executable as an independent computer program

(Doyle teaches a design that features a "list of applications," (column 15, line 14, Doyle). The list of applications allows an application to be selected to run along with the original program to provide an output from an input that the original program alone would be unable to provide as claimed. In addition, since the list of applications is initiated by the system, there must exist a service manager as claimed. However, such means are not expressly stated within Doyle's design.

Pratt discloses a design allowing a second program to be called to assist a first program to perform a desired task on data (column 3, lines 11-20, Pratt). Within the disclosure, Pratt states that the system resolves the reference to identify the script, the application program and the operation of the application program to execute the script and invoke the application program to execute the script to generate the result (column 3, lines 21-43, Pratt). Hence the claimed service manager must be present within the design since it's tasks are performed.

Both Doyle and Pratt teach systems allowing one program to call upon another program to perform an operation on a piece of data. Hence, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Doyle with those of Pratt, to provide a mechanism and a method for integrating first data created by a first application program and residing in a first data file or object belonging to the first application into a second data file or object (column 3, lines 11-15, Pratt)).

11. With regards to claim 31, Doyle teaches through Pratt a computer system

wherein the processing unit, responsive to the instructions, is further operable

for: arranging in the memory the at least one segment of computer code and a

data, comprising the at least one type of content that the computer program is

not configured to access, into a function-content group; and interfacing the

function-content group to the computer program to enable the computer program

to access the at least one type of content that the computer program is not

configured to access

     (Doyle teaches a design that features a "list of applications," (column 15,

line 14, Doyle). The list of applications allows an application to be selected to run

along with the original program to provide an output from an input that the

original program alone would be unable to provide, as claimed. No limitations

are placed as to the number of programs that may be run together. In addition,

since the list of applications is initiated by the system, there must exist a service

manager as claimed. However, such means are not expressly stated within

Doyle's design.

     Pratt discloses a design allowing a second program to be called to assist a

first program to perform a desired task on data (column 3, lines 11-20, Pratt).

Within the disclosure, Pratt states that the system resolves the reference to

identify the script, the application program and the operation of the application

program to execute the script and invoke the application program to execute the

script to generate the result (column 3, lines 21-43, Pratt). Hence the claimed

service manager must be present within the design since it's tasks are

performed.

Both Doyle and Pratt teach systems allowing one program to call upon

another program to perform an operation on a piece of data. Hence, it would

have been obvious to one skilled in the art, during the time of the invention, to

have combined the teachings of Doyle with those of Pratt, to provide a

mechanism and a method for integrating first data created by a first application

program and residing in a first data file or object belonging to the first application

into a second data file or object (column 3, lines 11-15, Pratt)).

12. With regards to claim 32, Doyle teaches through Pratt, the computer system

wherein the processing unit, responsive to the instructions, is operable for

identifying a plurality of segments of computer code by: locating at least two

segments of computer code from the plurality of segments of computer code that

each comprise a portion of computer code that indicates they can be executed by

the processing unit along with the computer program; and generating a list in the

memory comprising an identifier for each of the at least two segments of

computer code that indicates that the at least two segments of computer code

are available to be executed by the processing unit along with the computer

program

(Doyle teaches a design that features a "list of applications," (column 15, line 14, Doyle). The list of applications allows an application to be selected to run along with the original program to provide an output from an input that the original program alone would be unable to provide, as claimed. No limitation is given as to how many applications may be selected to run together, hence two or more may be selected as claimed. In addition, since the list of applications is initiated by the system, there must exist a service manager as claimed. However, such means are not expressly stated within Doyle's design.

Pratt discloses a design allowing a second program to be called to assist a first program to perform a desired task on data (column 3, lines 11-20, Pratt). Within the disclosure, Pratt states that the system resolves the reference to identify the script, the application program and the operation of the application program to execute the script and invoke the application program to execute the script to generate the result (column 3, lines 21-43, Pratt). Pratt's design also places no limitations on the number of applications that may be run along with the original program. Hence the claimed service manager must be present within the design since its tasks are performed.

Both Doyle and Pratt teach systems allowing one program to call upon another program to perform an operation on a piece of data. Hence, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Doyle with those of Pratt, to provide a mechanism and a method for integrating first data created by a first application

program and residing in a first data file or object belonging to the first application

into a second data file or object (column 3, lines 11-15, Pratt)).

13. With regards to claim 33, Doyle teaches through Pratt, a computer system

wherein the processing unit, responsive to the instructions, is operable for

selecting at least one segment of computer code by selecting at least two

segments of computer code from the plurality of segments of computer code

whose combined functionality will allow the computer program to access the at

least one type of content that the computer program is not configured to access

when the at least two segments of computer code are executed by the

processing unit along with the computer program

(Doyle teaches a design that features a "list of applications," (column 15,

line 14, Doyle). The list of applications allows an application to be selected to run

along with the original program to provide an output from an input that the

original program alone would be unable to provide, as claimed. No limitation is

given as to how many applications may be selected to run together, hence two or

more may be selected as claimed. In addition, since the list of applications is

initiated by the system, there must exist a service manager as claimed.

However, such means are not expressly stated within Doyle's design.

Pratt discloses a design allowing a second program to be called to assist a

first program to perform a desired task on data (column 3, lines 11-20, Pratt).

Within the disclosure, Pratt states that the system resolves the reference to

identify the script, the application program and the operation of the application

program to execute the script and invoke the application program to execute the

script to generate the result (column 3, lines 21-43, Pratt). Pratt's design also

places no limitations on the number of applications that may be run along with

the original program. Hence the claimed service manager must be present within

the design since its tasks are performed.

Both Doyle and Pratt teach systems allowing one program to call upon

another program to perform an operation on a piece of data. Hence, it would

have been obvious to one skilled in the art, during the time of the invention, to

have combined the teachings of Doyle with those of Pratt, to provide a

mechanism and a method for integrating first data created by a first application

program and residing in a first data file or object belonging to the first application

into a second data file or object (column 3, lines 11-15, Pratt)).

14. With regards to claim 34, Doyle teaches through Pratt, the computer system

wherein the processing unit, responsive to the instructions, is further operable for

configuring the at least two segments of computer code to be executed by the

processing unit along with the computer program in a particular order to allow the

computer program to access the at least one type of content that the computer

program is not configured to access

(Doyle teaches a design that features a "list of applications," (column 15,

line 14, Doyle). The list of applications allows an application to be selected to run

along with the original program to provide an output from an input that the

original program alone would be unable to provide, as claimed. No limitation is

given as to how many applications may be selected to run together, hence two or

more may be selected as claimed. Whenever multiple applications run together

to produce a single output, it is inherent that they will perform in order as claimed.

In addition, since the list of applications is initiated by the system, there must

exist a service manager as claimed. However, such means are not expressly

stated within Doyle's design.

Pratt discloses a design allowing a second program to be called to assist a

first program to perform a desired task on data (column 3, lines 11-20, Pratt).

Within the disclosure, Pratt states that the system resolves the reference to

identify the script, the application program and the operation of the application

program to execute the script and invoke the application program to execute the

script to generate the result (column 3, lines 21-43, Pratt). Pratt's design also

places no limitations on the number of applications that may be run along with

the original program. Hence the claimed service manager must be present within

the design since its tasks are performed.

Both Doyle and Pratt teach systems allowing one program to call upon

another program to perform an operation on a piece of data. Hence, it would

have been obvious to one skilled in the art, during the time of the invention, to

have combined the teachings of Doyle with those of Pratt, to provide a

mechanism and a method for integrating first data created by a first application

program and residing in a first data file or object belonging to the first application

into a second data file or object (column 3, lines 11-15, Pratt)).

15. With regards to claim 35, Doyle teaches through Pratt, the computer system,

wherein the processing unit, responsive to the instructions, is further operable for

configuring the at least two segments of computer code into a master-slave

relationship that causes the execution of one of the at least two segments of

computer code to be dependent on the execution of another of the at least two

segments of computer code

(Doyle discloses how applications can be started as child processes

(column 15, line 22, Doyle). When a child process exists, a parent process must

exist. This parent-child relationship is equivalent to a master-slave relationship.

In addition, since the list of applications is initiated by the system, there must

exist a service manager as claimed. However, such means are not expressly

stated within Doyle's design.

Pratt discloses a design allowing a second program to be called to assist a

first program to perform a desired task on data (column 3, lines 11-20, Pratt).

Within the disclosure, Pratt states that the system resolves the reference to

identify the script, the application program and the operation of the application

program to execute the script and invoke the application program to execute the

script to generate the result (column 3, lines 21-43, Pratt). Hence the claimed

service manager must be present within the design since it's tasks are

performed.

Both Doyle and Pratt teach systems allowing one program to call upon

another program to perform an operation on a piece of data. Hence, it would

have been obvious to one skilled in the art, during the time of the invention, to

have combined the teachings of Doyle with those of Pratt, to provide a

mechanism and a method for integrating first data created by a first application

program and residing in a first data file or object belonging to the first application

into a second data file or object (column 3, lines 11-15, Pratt)).

16. With regards to claim 36, Doyle teaches through Pratt, a computer-readable

medium having computer-executable instructions for accessing multiple types of

electronic content, comprising: logic for creating a list that comprises information

about a plurality of segments of computer code that can be executed along with a

computer program; logic for choosing at least one segment of computer code

from the plurality of segments of computer code, based on the information in the

list, that can be executed along with the computer program to process a type of

data that the computer program is not designed to process; logic to execute the

at least one segment of computer code along with the computer program in

response to an input to provide an output of the type of data that the computer

program is not designed to process, wherein a service manager connects the

computer program to at least one service container in response to the input,

wherein the plurality of segments of computer code and the at least one segment

of computer code are not executable as an independent computer program

(Doyle teaches a design that features a "list of applications," (column 15,

line 14, Doyle). The list of applications allows an application to be selected to run

along with the original program to provide an output from an input that the

original program alone would be unable to provide as claimed. In addition, since

the list of applications is initiated by the system, there must exist a service

manager as claimed. However, such means are not expressly stated within

Doyle's design.

Pratt discloses a design allowing a second program to be called to assist a

first program to perform a desired task on data (column 3, lines 11-20, Pratt).

Within the disclosure, Pratt states that the system resolves the reference to

identify the script, the application program and the operation of the application

program to execute the script and invoke the application program to execute the

script to generate the result (column 3, lines 21-43, Pratt). Hence the claimed

service manager must be present within the design since it's tasks are

performed.

Both Doyle and Pratt teach systems allowing one program to call upon

another program to perform an operation on a piece of data. Hence, it would

have been obvious to one skilled in the art, during the time of the invention, to

have combined the teachings of Doyle with those of Pratt, to provide a

mechanism and a method for integrating first data created by a first application

program and residing in a first data file or object belonging to the first application

into a second data file or object (column 3, lines 11-15, Pratt)).

17. With regards to claim 37, Doyle teaches through Pratt, he computer-readable

medium further comprising logic for choosing at least two segments of computer

code from the plurality of segments of computer code, based on the information

in the list, which can be executed along with the computer program to process a

type of data that the computer program is not designed to process

(Doyle teaches a design that features a "list of applications," (column 15,

line 14, Doyle). The list of applications allows an application to be selected to run

along with the original program to provide an output from an input that the

original program alone would be unable to provide as claimed. No limitation is

given as to how many applications may be selected to run together; hence two or

more may be selected claimed. In addition, since the list of applications is

initiated by the system, there must exist a service manager as claimed.

However, such means are not expressly stated within Doyle's design.

Pratt discloses a design allowing a second program to be called to assist a

first program to perform a desired task on data (column 3, lines 11-20, Pratt).

Within the disclosure, Pratt states that the system resolves the reference to

identify the script, the application program and the operation of the application

program to execute the script and invoke the application program to execute the

script to generate the result (column 3, lines 21-43, Pratt). Hence the claimed

service manager must be present within the design since it's tasks are

performed. Plus, no limitation is given as to how many applications may be

selected to run together; hence two or more may be selected claimed.

Both Doyle and Pratt teach systems allowing one program to call upon

another program to perform an operation on a piece of data. Hence, it would

have been obvious to one skilled in the art, during the time of the invention, to

have combined the teachings of Doyle with those of Pratt, to provide a

mechanism and a method for integrating first data created by a first application

program and residing in a first data file or object belonging to the first application

into a second data file or object (column 3, lines 11-15, Pratt)).

18. With regards to claim 38, Doyle teaches through Pratt, the computer-readable

medium, further comprising logic for linking the at least two segments of

computer code in a specific order of execution to provide a desired output of data

that the computer program is not designed to process when the at least two

segments of computer code are executed along with the computer program

(Doyle teaches a design that features a "list of applications," (column 15,

line 14, Doyle). The list of applications allows an application to be selected to run

along with the original program to provide an output from an input that the

original program alone would be unable to provide as claimed. No limitation is

given as to how many applications may be selected to run together; hence two or

more may be selected claimed. Whenever multiple applications run together to produce a single output, it is inherent that they will perform in order as claimed. In addition, since the list of applications is initiated by the system, there must exist a service manager as claimed. However, such means are not expressly stated within Doyle's design.

Pratt discloses a design allowing a second program to be called to assist a first program to perform a desired task on data (column 3, lines 11-20, Pratt). Within the disclosure, Pratt states that the system resolves the reference to identify the script, the application program and the operation of the application program to execute the script and invoke the application program to execute the script to generate the result (column 3, lines 21-43, Pratt). Hence the claimed service manager must be present within the design since it's tasks are performed. Plus, no limitation is given as to how many applications may be selected to run together; hence two or more may be selected claimed.

Both Doyle and Pratt teach systems allowing one program to call upon another program to perform an operation on a piece of data. Hence, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Doyle with those of Pratt, to provide a mechanism and a method for integrating first data created by a first application program and residing in a first data file or object belonging to the first application into a second data file or object (column 3, lines 11-15, Pratt)).

19. With regards to claim 39, Doyle teaches through Pratt, the computer-readable medium, wherein the logic for creating a list that comprises information about a plurality of segments of computer code comprises: logic for identifying at least two segments of computer code that each comprise a registration code that indicates that they can be executed along with the computer program; and logic for generating a list comprising an identification code for each of the at least two segments of computer code that indicates that the at least two segments of computer code are available to be executed along with the computer program

(Doyle teaches a design that features a "list of applications," (column 15, line 14, Doyle). The list of applications allows an application to be selected to run along with the original program to provide an output from an input that the original program alone would be unable to provide as claimed. No limitation is given as to how many applications may be selected to run together; hence two or more may be selected claimed. Whenever multiple applications run together to produce a single output, it is inherent that they will perform in order as claimed. Furthermore, Doyle describes a design that has the means by which to identify the application to be launched from the local user client machine (column 15, lines 18-21, Doyle). In addition, since the list of applications is initiated by the system, there must exist a service manager as claimed. However, such means are not expressly stated within Doyle's design.

Pratt discloses a design allowing a second program to be called to assist a first program to perform a desired task on data (column 3, lines 11-20, Pratt).

Within the disclosure, Pratt states that the system resolves the reference to identify the script, the application program and the operation of the application program to execute the script and invoke the application program to execute the script to generate the result (column 3, lines 21-43, Pratt). Hence the claimed service manager must be present within the design since it's tasks are performed. Plus, no limitation is given as to how many applications may be selected to run together; hence two or more may be selected claimed.

Both Doyle and Pratt teach systems allowing one program to call upon another program to perform an operation on a piece of data. Hence, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Doyle with those of Pratt, to provide a mechanism and a method for integrating first data created by a first application program and residing in a first data file or object belonging to the first application into a second data file or object (column 3, lines 11-15, Pratt)).

20. With regards to claim 40, Doyle teaches through Pratt, computer-readable medium, further comprising logic for arranging the at least one segment of computer code and a data element, comprising the type of data that the computer program is not designed to process, into a function-data group; and logic for interfacing the function-data group to the computer program to enable the computer program to provide the output of the type of data that the computer program is not designed to process

(Doyle teaches a design that features a "list of applications," (column 15,

line 14, Doyle). The list of applications allows an application to be selected to run

along with the original program to provide an output from an input that the

original program alone would be unable to provide as claimed. No limitation is

given as to how many applications may be selected to run together; hence two or

more may be selected claimed. Whenever multiple applications run together to

produce a single output, it is inherent that they will perform in order as claimed.

Furthermore, for data to be processed, it typically is grouped together, for

instance within a data structure of some form. In addition, since the list of

applications is initiated by the system, there must exist a service manager as

claimed. However, such means are not expressly stated within Doyle's design.

Pratt discloses a design allowing a second program to be called to assist a

first program to perform a desired task on data (column 3, lines 11-20, Pratt).

Within the disclosure, Pratt states that the system resolves the reference to

identify the script, the application program and the operation of the application

program to execute the script and invoke the application program to execute the

script to generate the result (column 3, lines 21-43, Pratt). Hence the claimed

service manager must be present within the design since it's tasks are

performed. Plus, no limitation is given as to how many applications may be

selected to run together; hence two or more may be selected claimed.

Both Doyle and Pratt teach systems allowing one program to call upon

another program to perform an operation on a piece of data. Hence, it would

have been obvious to one skilled in the art, during the time of the invention, to

have combined the teachings of Doyle with those of Pratt, to provide a

mechanism and a method for integrating first data created by a first application

program and residing in a first data file or object belonging to the first application

into a second data file or object (column 3, lines 11-15, Pratt)).

21. With regards to claims 41, 44 and 47, Doyle teaches through Pratt, the method

wherein each service container comprises a data object, a code object, and a

loader identification

(Doyle teaches a design that features a "list of applications," (column 15,

line 14, Doyle). The list of applications allows an application to be selected to run

along with the original program to provide an output from an input that the

original program alone would be unable to provide as claimed. In addition, since

the list of applications is initiated by the system, there must exist a service

manager along with reference pointers as claimed. However, such means are

not expressly stated within Doyle's design.

Pratt discloses a design allowing a second program to be called to assist a

first program to perform a desired task on data (column 3, lines 11-20, Pratt).

Within the disclosure, Pratt states that the system resolves the reference to

identify the script, the application program and the operation of the application

program to execute the script and invoke the application program to execute the

script to generate the result (column 3, lines 21-43, Pratt). Hence the claimed

service manager must be present within the design since it's tasks are

performed. Plus, Pratt's design allows for references to scripts, scripts which are

in objects, to exist (column 3, lines 21-43, Pratt). Hence, means are present

within Pratt's design for the claimed traits.

Both Doyle and Pratt teach systems allowing one program to call upon

another program to perform an operation on a piece of data. Hence, it would

have been obvious to one skilled in the art, during the time of the invention, to

have combined the teachings of Doyle with those of Pratt, to provide a

mechanism and a method for integrating first data created by a first application

program and residing in a first data file or object belonging to the first application

into a second data file or object (column 3, lines 11-15, Pratt)).

22. With regards to claims 42, 45 and 48, Doyle teaches through Pratt, the method

wherein each code object references at least one service object

(Doyle teaches a design that features a "list of applications," (column 15,

line 14, Doyle). The list of applications allows an application to be selected to run

along with the original program to provide an output from an input that the

original program alone would be unable to provide as claimed. In addition, since

the list of applications is initiated by the system, there must exist a service

manager along with reference pointers as claimed. However, such means are

not expressly stated within Doyle's design.

Pratt discloses a design allowing a second program to be called to assist a

first program to perform a desired task on data (column 3, lines 11-20, Pratt).

Within the disclosure, Pratt states that the system resolves the reference to

identify the script, the application program and the operation of the application

program to execute the script and invoke the application program to execute the

script to generate the result (column 3, lines 21-43, Pratt). Hence the claimed

service manager must be present within the design since it's tasks are

performed. Plus, Pratt's design allows for references to scripts, scripts which are

in objects, to exist (column 3, lines 21-43, Pratt). Hence, means are present

within Pratt's design for the claimed traits.

Both Doyle and Pratt teach systems allowing one program to call upon

another program to perform an operation on a piece of data. Hence, it would

have been obvious to one skilled in the art, during the time of the invention, to

have combined the teachings of Doyle with those of Pratt, to provide a

mechanism and a method for integrating first data created by a first application

program and residing in a first data file or object belonging to the first application

into a second data file or object (column 3, lines 11-15, Pratt)).

23. With regards to claims 43, 46 and 49, Doyle teaches through Pratt, a method

wherein each service object is stored in a cache, separate from each service

container

(Doyle teaches a design that features a "list of applications," (column 15, line 14, Doyle). The list of applications allows an application to be selected to run along with the original program to provide an output from an input that the original program alone would be unable to provide as claimed. In addition, since the list of applications is initiated by the system, there must exist a service manager along with reference pointers as claimed. However, such means are not expressly stated within Doyle's design.

Pratt discloses a design allowing a second program to be called to assist a first program to perform a desired task on data (column 3, lines 11-20, Pratt). Within the disclosure, Pratt states that the system resolves the reference to identify the script, the application program and the operation of the application program to execute the script and invoke the application program to execute the script to generate the result (column 3, lines 21-43, Pratt). Hence the claimed service manager must be present within the design since its tasks are performed. Plus, Pratt's design allows for references to scripts, scripts which are in objects, to exist (column 3, lines 21-43, Pratt). The cache-storing step claimed is simply caching in temporary storage such as RAM. Pratt's design makes use of computers and hence the caching technique exists.

Both Doyle and Pratt teach systems allowing one program to call upon another program to perform an operation on a piece of data. Hence, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Doyle with those of Pratt, to provide a

mechanism and a method for integrating first data created by a first application

program and residing in a first data file or object belonging to the first application

into a second data file or object (column 3, lines 11-15, Pratt)).

24. With regards to claim 50, Doyle teaches through Pratt, a method for accessing;

multiple types of electronic content from a program, comprising the steps of:

defining a plurality of service containers accessible to the program, each service

container corresponding to a specific function, each service container including:

a data object, such that each data object includes data required to enable the

specific function corresponding to the service container to be achieved; a code

object, each code object referencing at least one segment of programming code

stored separately from the service container, such that the at least one segment

of programming code referenced by the code object includes programming code

required to enable the specific function corresponding to the service container to

be achieved; and a loader identification, the loader identification providing the

program with information required to load the segment of programming code

referenced by the code object; requesting an input to be processed to obtain an

output; identifying one of the plurality of service containers whose corresponding

specific function is able to process the input; parsing the service container thus

identified, to determine: data required to enable the specific function

corresponding to the service container to be achieved; an identity of the at least

one segment of programming code required to enable the specific function

corresponding to the service container to be achieved; and information required

to load each segment of programming code referenced by the code object;

retrieving the at least one segment of programming code required to enable the

specific function corresponding to the service, container to be achieved; and

executing the at least one segment of computer code under the control of the

computer program to process the input and obtain the output

(Doyle teaches a design that features a "list of applications," (column 15,

line 14, Doyle). The list of applications allows an application to be selected to run

along with the original program to provide an output from an input that the

original program alone would be unable to provide as claimed. In addition, since

the list of applications is initiated by the system, there must exist a service

manager as claimed. However, such means are not expressly stated within

Doyle's design.

Pratt discloses a design allowing a second program to be called to assist a

first program to perform a desired task on data (column 3, lines 11-20, Pratt).

Within the disclosure, Pratt states that the system resolves the reference to

identify the script, the application program and the operation of the application

program to execute the script and invoke the application program to execute the

script to generate the result (column 3, lines 21-43, Pratt). Hence the claimed

service manager must be present within the design since it's tasks are

performed.

Both Doyle and Pratt teach systems allowing one program to call upon another program to perform an operation on a piece of data. Hence, it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Doyle with those of Pratt, to provide a mechanism and a method for integrating first data created by a first application program and residing in a first data file or object belonging to the first application into a second data file or object (column 3, lines 11-15, Pratt)).

Remarks

The examiner has carefully reviewed the amended claims carefully along with the attached remarks. In addition, a new search has been carried out. It is from this search that a new art has been discovered and it has been used in combination with the Doyle art previously presented. It is believed that the combined references will point out to the applicant and their representatives, that the claimed design unfortunately lacks novelty. Should any other unique traits be present for the claimed invention, it is encouraged that the applicant and their representatives amend the claims to reflect such changes.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Azizul Choudhury whose telephone number is (571) 272-3909. The examiner can normally be reached on M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Jack Harvey can be reached on (571) 272-3896. The fax phone number for

the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the

Patent Application Information Retrieval (PAIR) system. Status information for

published applications may be obtained from either Private PAIR or Public PAIR.

Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see http://pair-direct.uspto.gov. Should

you have questions on access to the Private PAIR system, contact the Electronic

Business Center (EBC) at 866-217-9197 (toll-free).

JACK B. HARVEY
SUPERVISORY PATENT EXAMINER

AC